



Universitat
Autònoma
de Barcelona



Creación de una radio musical participativa con difusión online y FM

Memoria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Alejandro de Quintana Schmidt
i dirigit per
Josep Maria Ganyet
Bellaterra, 17 de septiembre de 2008

Proyecto: Creación de una radio musical participativa con difusión online y FM
Por Alejandro de Quintana

XXXXXXXX

Agradecimientos:

*A los profesores que he tenido durante la carrera, por todo lo que
me han enseñado*

A mis amigos, por estar ahí

A merc por sus sabios consejos y paciencia

INDICE

1. Introducción.....	6
1.1. Objetivos.....	6
1.2. Plan de trabajo.....	8
1.3. Análisis de requerimientos	9
1.3.1. Funcionales	9
1.3.2 No Funcionales.....	10
2. Diseño.....	11
2.1 Diseño base de datos	11
2.2. Casos de uso:.....	17
2.3 Modificación en la Base de datos tras el estudio de los casos de uso	20
2.4. Esqueleto interfaz de la emisora:	22
2.5. Modificación en la Base de datos tras el primer diseño de la interfaz	24
3. Algoritmo de decisión	26
3.1. Sistema de representación	26
3.2. Estudio funcionalidad del Algoritmo	31
3.2.1. Sistema de coordenadas	33
3.2.2. Peso de las características:	33
3.2.3. Factor de novedad.....	34
3.2.4. Calculo de la distancia	35
3.3. Algoritmo.....	37
3.4. Modificación en la Base de datos tras la selección del algoritmo de decisión	39
4. Pruebas realizadas.....	42
4.1. Modificaciones en la base de datos tras las pruebas	44
5. Conclusiones.....	47
5.1. Objetivos conseguidos y Temas pendientes	47
5.2. Futuras mejoras	47
6. Herramientas utilizadas	50
7. Bibliografía.....	51
Anexos	52
Modelo Físico de la Base de Datos.....	52
Ejemplo del funcionamiento del algoritmo paso a paso	56

1. Introducción

El modelo clásico de la red esta cambiando. Los usuarios, antiguamente simples observadores y lectores del contenido, ya no se conforman con serlo y quieren participar de manera activa. Primero fue a través de programas de comunicación (al estilo del extremadamente popular Messenger), y foros, para ahora, ya expresar sus opiniones vía blogs, exponer sus gustos o aficiones a través de paginas (al estilo de videos de Youtube o redes sociales como Facebook) o definir el estilo de un servicio para ajustarlo a sus gustos

Este proyecto se basa en una de las tendencias más importantes de la actualidad, las emisoras de radios personalizadas. En ellas los usuarios definen los temas y/o artistas que les gustan, para crear una selección de temas musicales a su gusto.

Así este proyecto plantea la creación de una radio que emita tanto por Internet como vía Web. Se crea, a través de una pagina Web, una interfaz que permita a cualquier usuario interactuar con la emisora, tanto de manera pasiva (a través de la consulta de información de la emisora), como de manera activa, ya sea votando a los temas que están sonando como proponiendo nuevos candidatos para la emisora

A partir de la información facilitada por los usuarios, la emisora emitirá unos temas o otros, nunca sobre una lista establecida, y manteniéndose siempre en un estilo de emisora

Así pues, se ha decidido implementar una radio de rock, que siga estas guías maestras

1.1. Objetivos

El principal objetivo de este proyecto es la creación de un sistema de difusión a través de Internet y FM, con capacidad de decisión sobre los temas que sonaran basada en las aportaciones de gustos de los usuarios

La parte más importante a la hora de crear este proyecto es el diseño de un algoritmo de decisión. El algoritmo ha de poder gestionar la selección del siguiente tema, a partir de la información que recibirá de los usuarios respecto a si valoran positivamente o no, los temas que están sonando.

El algoritmo no ha de seleccionar solo la canción mas votada, sino a partir de una base de candidatos posibles, escoger aquella canción que mas se adecue a las preferencias definidas por los usuarios

El método para poder relacionar los diferentes temas consiste en utilizar un sistema de clasificación de los mismos a partir de una serie de características; de esta manera cuando un usuario este votando lo que esta votando es si le gustan (o no) las características, no al tema en si.

Cuantas más características tengamos de los temas, y por tanto trabajemos con ellas, más precisión y potencia tendrá el algoritmo. Siempre y cuando estas características estén bien definidas, por supuesto, ya que una mala clasificación seria contraproducente. Y dentro del mundo musical esto no es fácil, resulta extremadamente complejo encontrar una clasificación única de los temas y grupos.

Teniendo en cuenta estas consideraciones se ha creado una radio, con una interfaz sencilla, para que los oyentes puedan expresar sus opiniones y gustos en base a los votos que emitan

1.2. Plan de trabajo

Este proyecto esta basado en tres aspectos: Una base de datos, una interfaz y un algoritmo de decisión

Dado que estos tres aspectos están interrelacionados, y la modificación de uno, puede afectar al resto, se ha establecido el siguiente plan de trabajo

Primera etapa:

Diseño inicial de la base de datos

Selección de un sistema de representación para el algoritmo

Segunda etapa:

Definir casos de uso

Ajuste de la base de datos, para adecuarla

Creación de un modelo de la interfaz

Tercera etapa:

Creación de un algoritmo de decisión

Ajuste de la base de datos

Cuarta etapa:

Pruebas de funcionamiento y ajustes necesarios

1.3. Análisis de requerimientos

1.3.1. Funcionales

Generales

- Emitir a través de Internet y FM
- Escoger, a través del algoritmo de decisión, cual es la siguiente canción que sonara a partir de los votos de los usuarios
- Si no hay ninguna canción candidata, escoger una canción aleatoria
- Control de los errores en la gestión con la base de datos
- Control de no repetición de temas

Usuarios no registrados

- Escuchar la radio
- Buscar por temas o artista

Usuarios registrados

- Proponer canciones
- Votar un tema que esta sonando

Administrador

- Cambiar la configuración del sistema
- Introducir o modificar :
 - Características de los temas
 - Temas
 - Artistas
 - Influencias
 - Datos usuarios
- Validar peticiones de los usuarios

- Eliminar de manera lógica temas recurrentes o no votados en absoluto
- Ver logs de votación, emisión y cambios en la configuración
- Activar y desactivar la radio

1.3.2 No Funcionales

- Capacidad de emisión vía FM
- Conexión a Internet o conexión local
- Sistema fácil de mantener y/o ampliar
- Limitación en la interactividad debido a la velocidad de conexión de los usuarios
- Limitación sobre la cantidad de usuarios que pueden escuchar la radio a la vez, así como la calidad de emisión (tanto de carácter técnico como económico)

2. Diseño

2.1 Diseño base de datos

Una de las piezas fundamentales del proyecto es la base de datos, que nos permitirá almacenar, consultar y modificar los datos que necesitaremos para el funcionamiento de nuestra emisora de radio.

El primer diseño que se plantea es, en líneas generales, una estructura básica sobre la cual se realizarán modificaciones futuras, adaptándolo a los diferentes requerimientos o problemas que nos encontremos

Este primer diseño se basa en un concepto principal: definir la piedra angular para la reproducir en nuestra radio

En la actualidad existen varias emisoras, (personales o de carácter genérico) que emiten su contenido a través de la red. Cada una de ellas se basa en un sistema para la emisión.

Así por un lado tenemos emisoras del estilo de Pandora que basan toda su concepción en los temas que emiten. Estas emisoras, priorizan que el usuario introduzca una canción que sirva como base para la emisión, y a partir de ella sacar la información para el resto de la emisión que se realizara.

Dentro de este grupo, donde se da la prioridad a los temas, hay que destacar un grupo de emisoras: por un lado Musicoverly, y más recientemente Rockola.fm, las cuales proponen que sea el propio usuario el que introduzca ya unas características de un tema sobre los que basar las futuras emisiones

Por otro lado tenemos el caso de emisoras como Last.fm, donde el usuario no entra un tema, sino lo que entra es el nombre de un grupo que servirá como base para las futuras selecciones.

Así nuestra decisión, es si basamos la emisión en temas, o bien en artistas

En la actualidad podemos ver que la tendencia del mundo musical esta apuntando en dar cada vez mas importancia a los temas. Es cada vez mas común que la gente consuma a través de plataformas como iTunes o eBay a la compra de temas concretos, relegando cada vez mas los clásicos álbumes, siendo lo mas común que los álbumes mas vendidos sean o bien recopilatorios de un artista (siempre canciones conocidos, con la salvedad de una o dos temas nuevos, que pasan mas o menos inadvertidos, ya que los compradores están interesados en los clásicos del artista) o por otro lado un popurrí de canciones de éxito, que las emisoras, anuncios o series han hecho populares.

Y dado que nuestro objetivo es la creación de una emisora, y por tanto la opinión de los oyentes/usuarios es importante, se ha optado por escoger como base del sistema los temas

Una vez tomada la base, se crea la primera BD, que como puede verse [ver Figura 1] es bastante simple, y simplemente cumple el objetivo de ser un esbozo sobre el cual estructurar la base del proyecto.

Se puede observar que la Base de Datos consta de 5 Tablas,

Las columnas que se han declarado tienen tres funciones: interrelación entre las tablas, campos que creamos obligatorios para la ejecución (votos,...) e informativos. Estos últimos son campos que contienen información que facilitara las búsquedas, así como la información que se presentara por pantalla al usuario

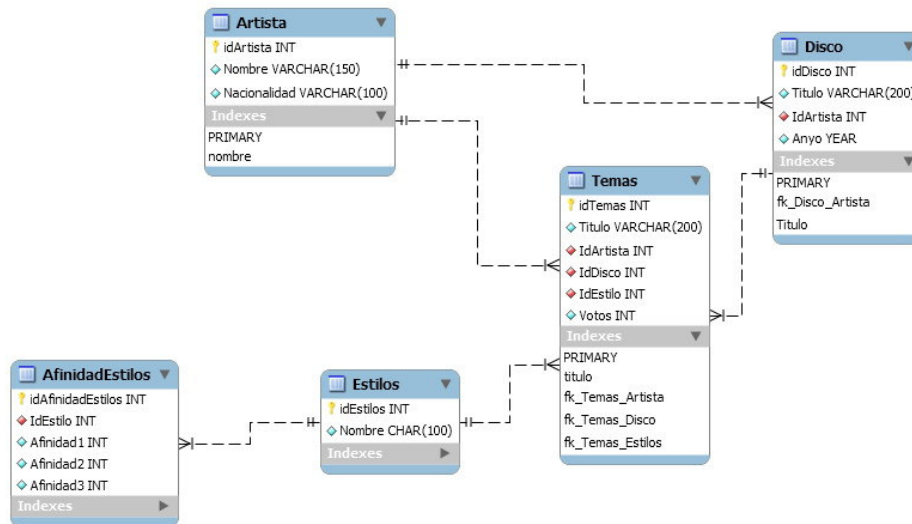


Figura 1: Primer diseño de la base de datos

Se han usado dos engines en la creación de las tablas:

Por un lado tenemos las tablas: Artista, Disco, Estilos y AfinidadEstilos. Todas ellas usan un engine MyIsam. Las tablas MyIsam, son especialmente indicadas dado su alta velocidad y facilidad de ser cacheadas [tal y como se comenta en la documentación de mysql]

Las tablas que hemos comentado, no se verán modificadas en tiempo real, ya que su objetivo es básicamente de consulta, por lo que nos interesa que sean lo mas rápidas posibles. Además, estas tablas no serán nunca modificadas por los usuarios, sino solamente por el administrador, por lo que el tiempo de bloqueos, y posibles conflictos será menor que si también fueran modificadas por el resto de usuarios

En cambio, la tabla Temas, difiere de las comentadas hasta ahora. Por un lado se puede escoger un engine MyIsam para darle el máximo de rapidez posible y facilitar las consultas que quieran realizar los usuarios, pero por otro, se ha de pensar un tipo engine que mejor nos permita gestionar múltiples usuarios que interactúen con la tabla a través de las modificaciones/inserciones

que ejecuten. Al final se ha decidido escoger este aspecto como principal, por tanto la tabla Temas, usa un engine InnoDB

En lo referente a las columnas de las tablas, cabria destacar algunos detalles:

- En la tabla Artistas: ``Nombre` VARCHAR(150) NOT NULL ,`

Aunque usar columnas de tipo char, y por tanto tablas mylsam estáticas, facilita la recuperación, así como reduce la complejidad, y mantenimiento (al evitar el uso de OPTIMIZE TABLE para mantener el máximo rendimiento) se ha optado por usar valores varchar, ya que se desconoce el valor exacto del tamaño del campo. Se espera poder optimizar los resultados y el tipo de la columna a través del uso, y los datos que se vayan introduciendo.

Además esta columna, como otras (como “titulo” en la tabla disco) no se permite valores nulos, ya que se considera que la información es importante para el buen funcionamiento de la aplicación

- En la tabla temas: Se ha declarado un valor inicial a la columna votos. [Ej: ``Votos` INT NOT NULL DEFAULT 0`]
- En todas las tablas: Todas las claves primarias de las diferentes tables se han declarado de tipo autoincremental [Ej: ``idAfinidadEstilos` INT NOT NULL AUTO_INCREMENT]`

Una vez creada este primer diseño de la Base, se deben analizar los posibles problemas, así como consideraciones que no se han tenido en cuenta, para poder modificar y hacerla mas adecuada al problema planteado

Después de analizarlo, comprobamos que existen los siguientes errores:

- Un tema puede ser tocado por varios artistas. En la actualidad, no contemplamos esta situación, por lo que un artista estaría varias veces en la tabla de artista, una en solitario, y otras tantas como

duetos o colaboraciones hubiera realizado. Esto es un gasto inútil de espacio en la tabla.

- Un tema no es exclusivo de un álbum. Si bien es cierto que un tema, ha salido originalmente en un disco, puede haber salido en más de una recopilación, ya sea del artista como genéricos. Y son muchos, como ya hemos comentado anteriormente, los usuarios que se guían por un recopilatorio, mas o menos emblemático o actual como guía para escuchar una canción
- Otro aspecto ha mejorar es introducir las características que se han decidido en el sistema de representación, para el algoritmo de decisión [la explicación detallada de la selección de características esta explicada en la **sección 3.1**]

Del sistema de caracterizas definido, deducimos que no se ha definido el ambiente de los temas, ni la influencia de los artistas entre si.

Para poder dar solución a estos problemas se ha procedido a los siguientes cambios en la base de datos:

- Problema de una tema – varios artistas: se ha creado una nueva tabla, Interpretado, la esta compuesta por los Ids tanto de temas como de Artista (su clave primaria es la combinación de los dos). De esta manera se pueden tener tantas entradas como se quieran de un tema, ya que cada entrada esta relacionado con un artista. No se ha creado un identificador único de interpretado y después la relación de un tema, y N artista, ya que no existe un máximo establecido, y podría comportar dificultades futuras (como es el caso de alguna canción navideña cantada por múltiples artistas)
- Problema un tema – varios discos: Al igual que con el caso anterior se ha creado una nueva tabla, Perteneciente, compuesta

por el IdTema y IdDisco, donde la clave primaria es la combinación de los dos.

Dado que la información de relación de los temas, con los artistas y discos, ha sido trasladada a estas nuevas tablas “intermedias”, se ha eliminado las columnas de idArtista y IdDisco de la tabla de Temas.

También se ha creado una nueva tabla intermedia que relaciona los artistas con los discos. Todas estas nuevas tablas se han creado con el engine Mylsam, ya que se tratan de tablas de consulta, y por tanto, solo el administrador podrá modificarlas.

Y en lo referente al sistema de representación, se crea dos nuevas tablas para definir las nuevas características (afinidad e influencia entre grupos), y una tabla de interrelación para ambientes. Las tablas de Ambiente, y AfinidadAmbiente siguen el mismo estilo que tenían Estilo y AfinidadEstilo,

Así nuestra Base de Datos quedara de la siguiente manera:

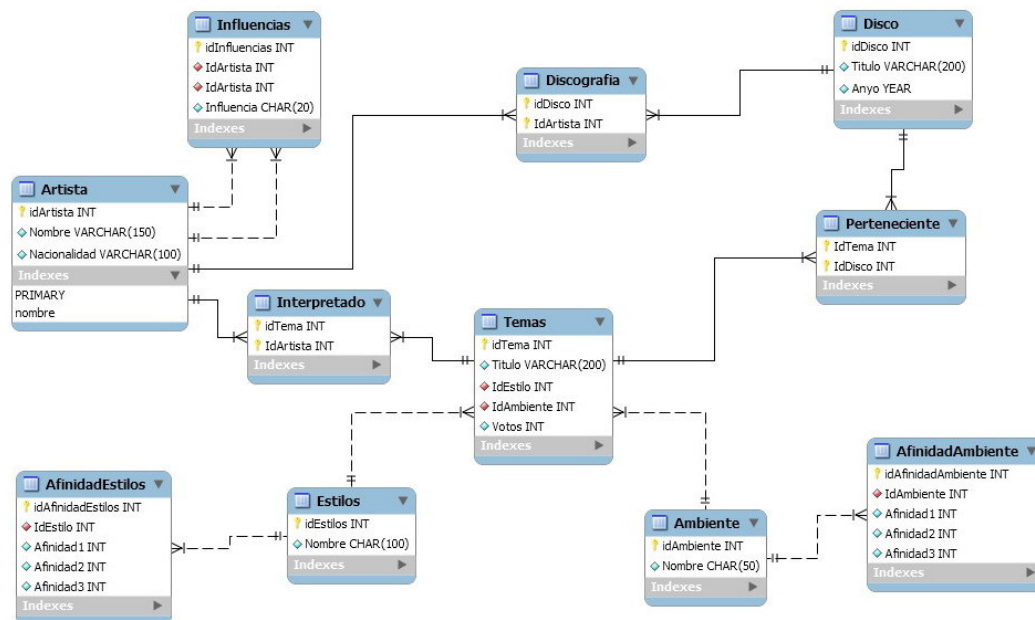


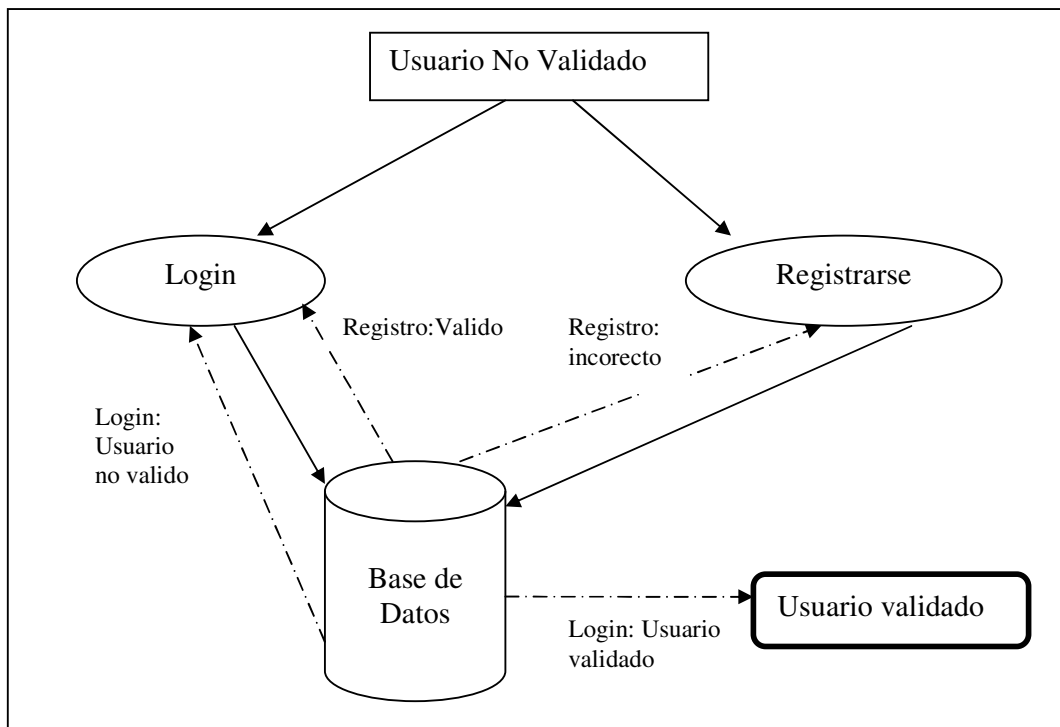
Figura 2: Modificaciones al primer diseño de la base de datos

2.2. Casos de uso:

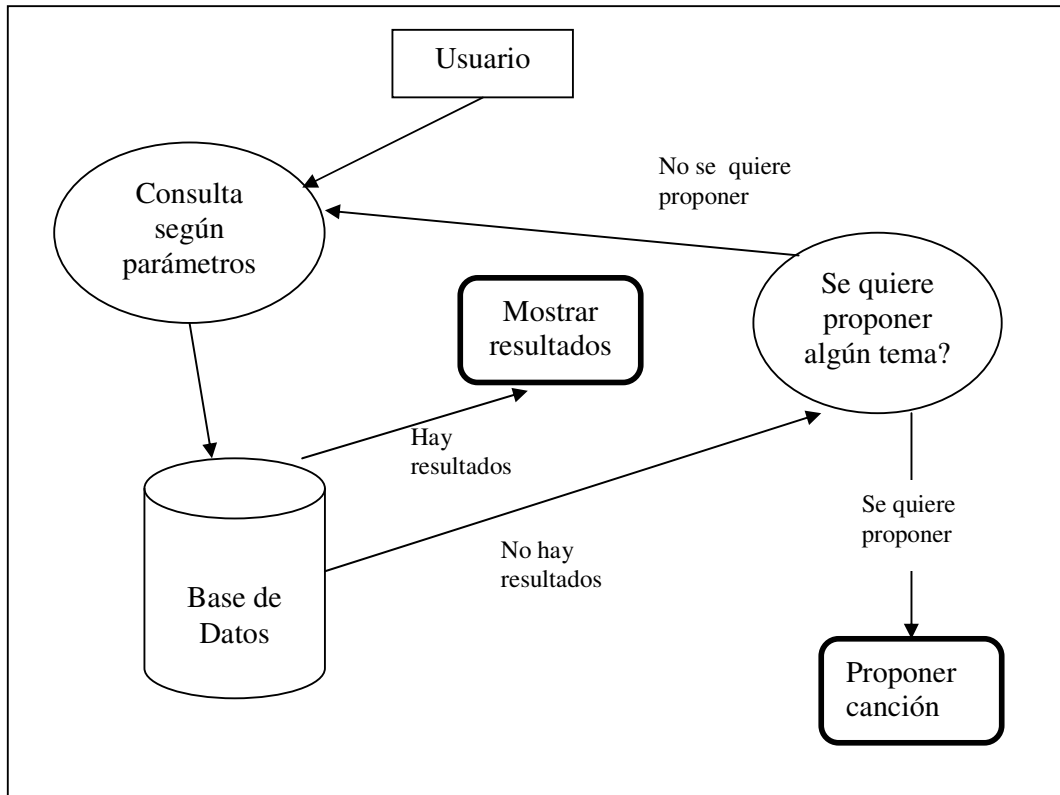
“Un caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización de software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico” [Definición de Caso de Uso de la Wikipedia]

Con el análisis de requerimientos definido, se crean los diferentes casos de uso, para analizar las diferentes opciones de interacción entre los diferentes usuarios y el sistema

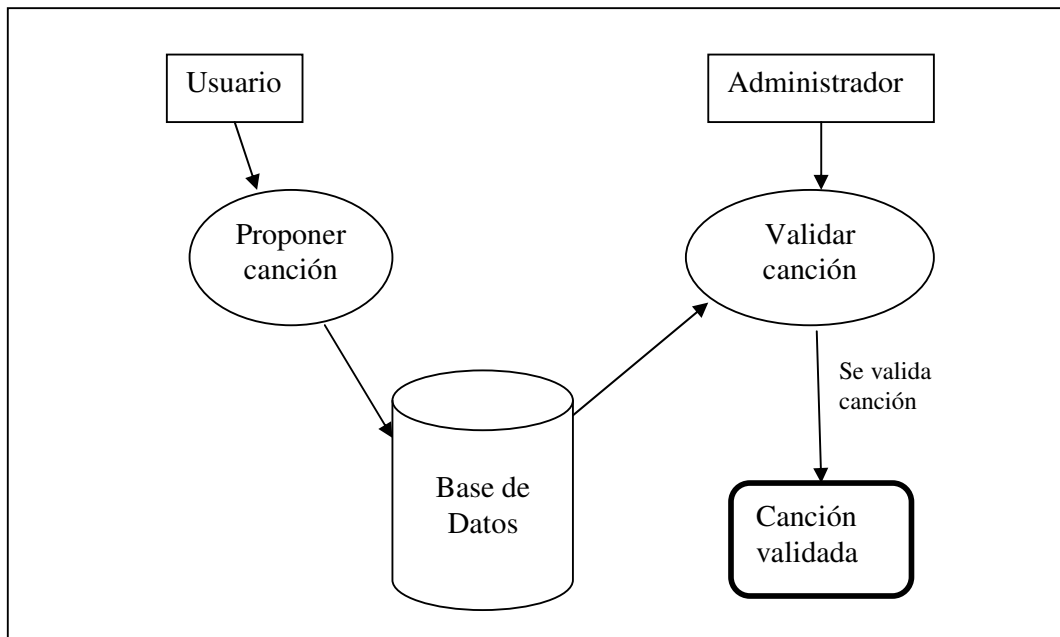
Validar/Registrar usuario:



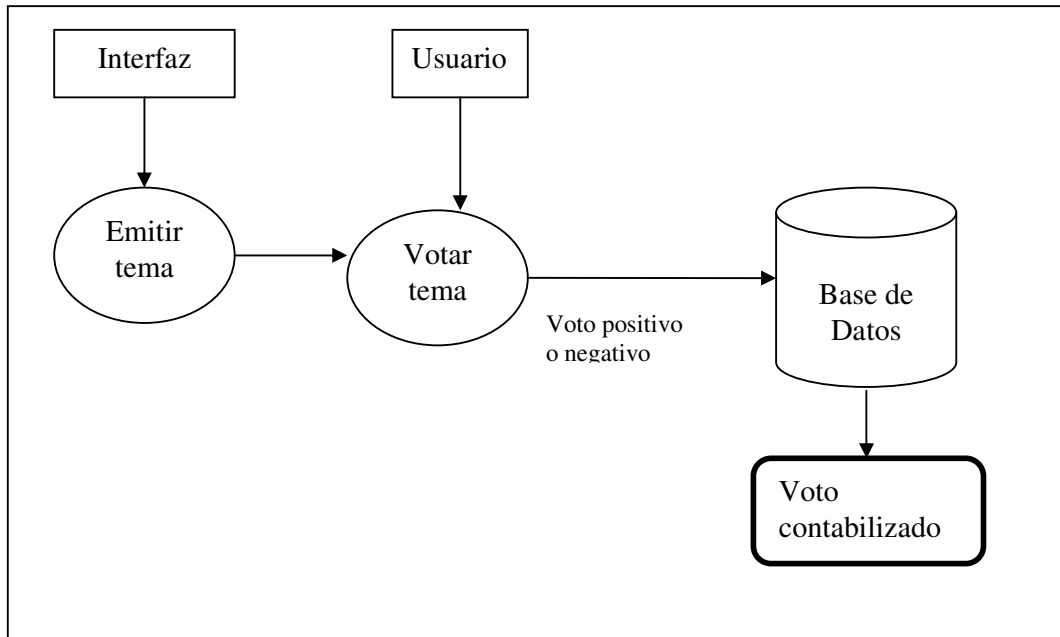
Consultar:



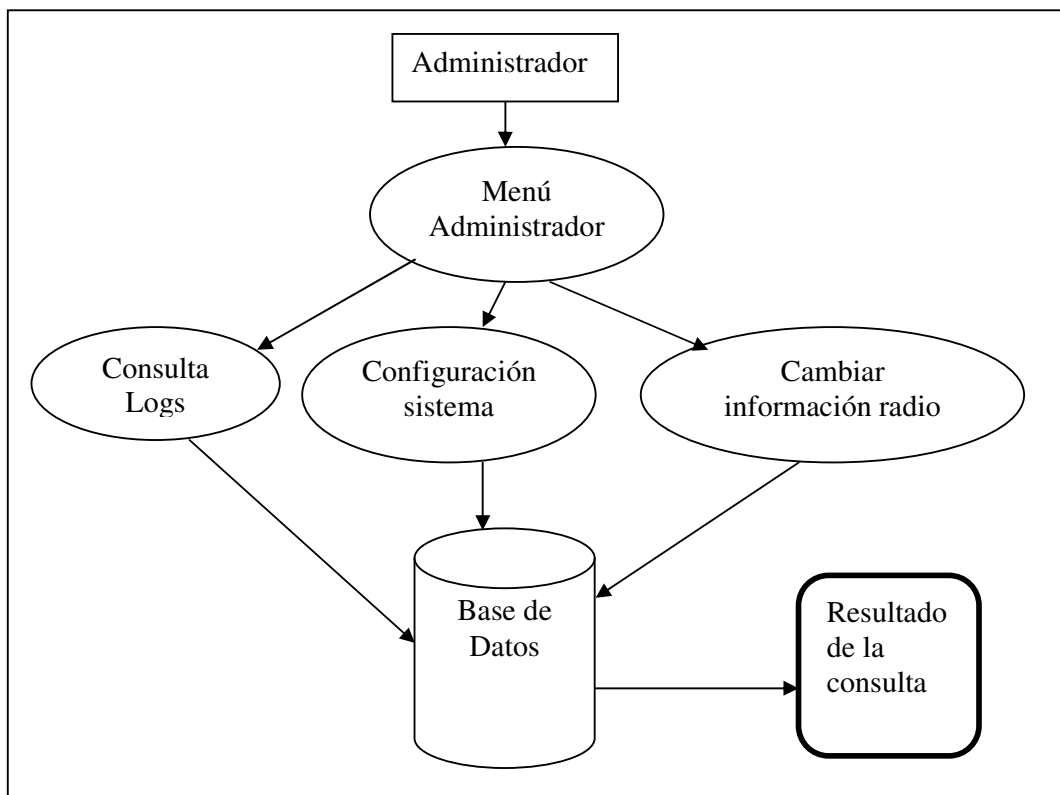
Proponer canción



Votar canción



Administración



2.3 Modificación en la Base de datos tras el estudio de los casos de uso

Tras haber hecho el diseño de los casos de uso, se revisara el diseño de la base de datos, para ver si esta cumple con los casos que se han definido

Después del análisis, vemos que hay ciertas características que no se han tenido en cuenta. Las podemos clasificar en dos aspectos: Características de funcionalidad y características de información

Las características de información corresponden a aquellas tablas o columnas, que proporcionan mas información al usuario, bien para facilitar búsquedas de información o bien información que se mostrara al usuario por pantalla

Dentro de este grupo tenemos lo siguiente:

- La creación de la tabla de grupos (que se puede también usar en el apartado de funcionalidad, al aportar unas posibles nuevas características a estudiar de cada tema)
- La inclusión de campos en la tabla de temas de las columnas: "Anysalida", "Duracion", "Letra", que nos aportaran tanto información para búsquedas, como información a mostrar por pantalla

Las características funcionales, son aquellos campos o tablas, que nos permitirán añadir funcionalidades al proyecto (algunas que no se habían tenido en cuentas y otras que han surgido mientras se llevaba a cabo). En este caso, cabe considerar las siguientes funcionalidades:

- Funcionalidad de Voto: Se añaden los siguientes campos a la tabla de temas: "vplus", "vminus", y "temporis". Los dos primeros recogen los votos de los usuarios a favor y en contra del tema que esta sonando, y el último temporis, es la columna que nos permite

controlar que una canción no se repita cierto tiempo después de ser emitida

- Funcionalidad de Peticiones de los usuarios: Se añade la tabla Peticiones, que recoge las propuestas de los usuarios. La tabla incluye el título del tema, así como el artista, y la fecha de la propuesta. El ultimo campo definido, “idtema”, empezara con un valor por defecto de null. En el momento que la canción se valida, se actualizara para vincularla al idtema correspondiente. De esta manera se tendrá un control de que canciones han sido ya validadas y cuales no

Después de estos cambios la base de datos queda de la siguiente manera

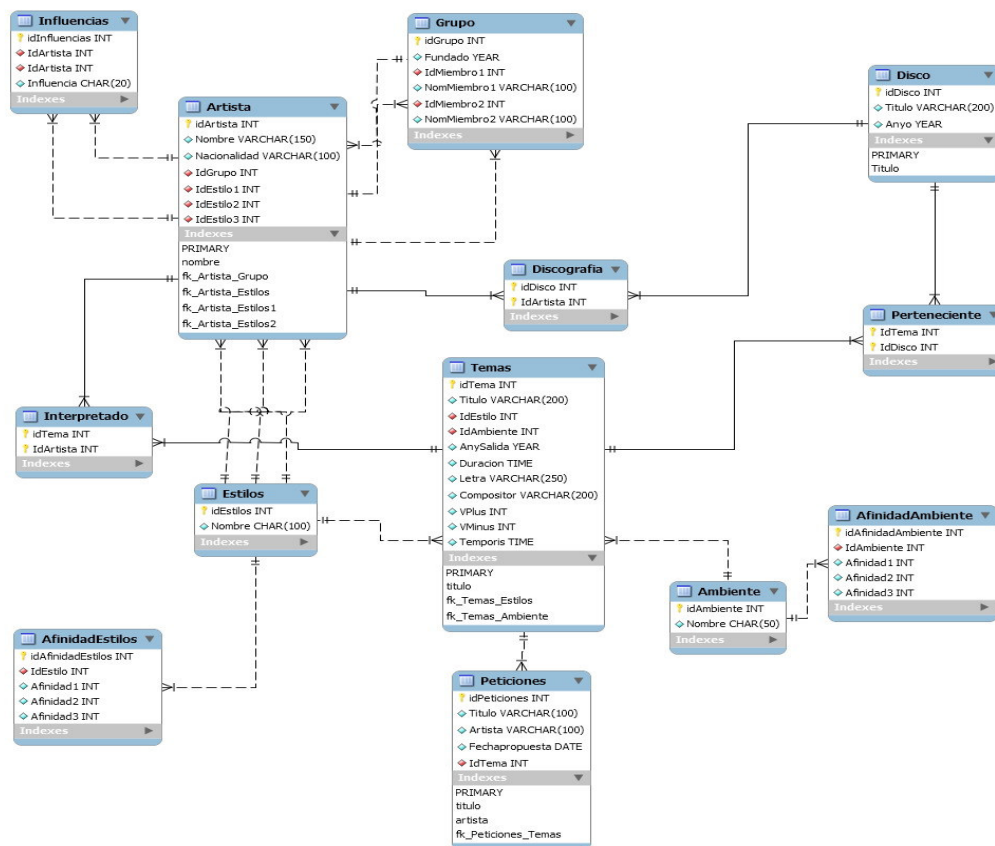


Figura 3: Base de Datos tras el estudio de casos de uso

2.4. Esqueleto interfaz de la emisora:

Una vez definidos los casos de uso y actualizado la base de datos, creamos una primera implementación, de la estructura de la interfaz final

La estructura de nuestra aplicación seguirá un modelo MVC, con lo que se deja la lógica de control para un momento posterior ya que esta estructura solo hace referencia a la vista.

Se crean 2 páginas, correspondientes a la vista del usuario, y la vista del administrador

La vista del usuario:

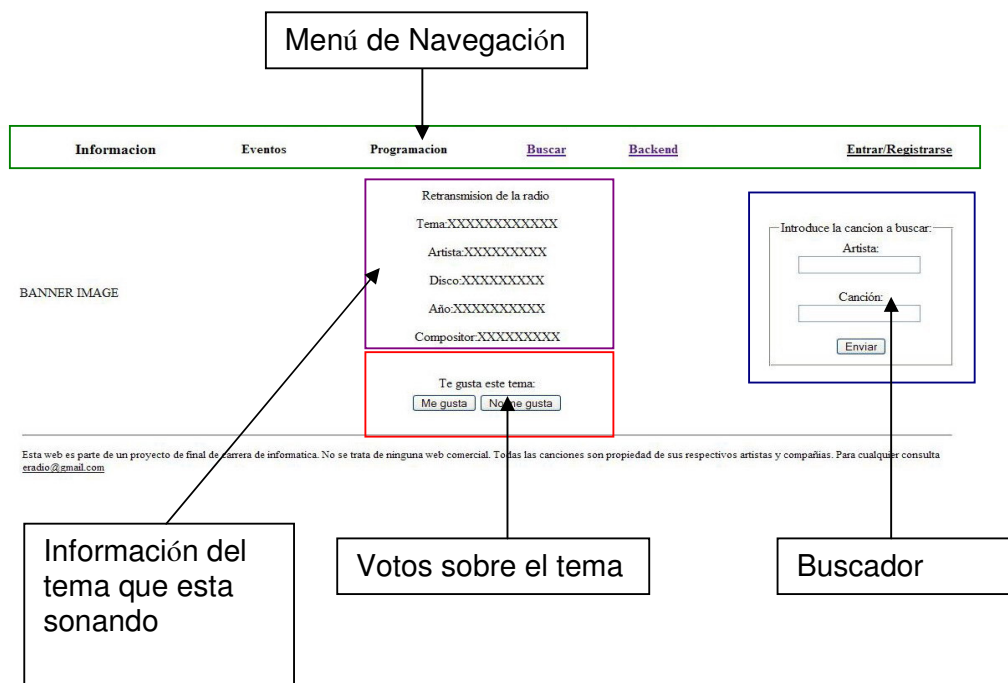


Figura 4: Vista de la interfaz del usuario

En este momento del proyecto, no importa tanto la información, sino cómo ésta será presentada, es decir si la interfaz nos permite todas las funciones que hemos planteado en el caso de uso

En el menú de Navegación, tenemos las opciones de información genérica de la emisora, así como la opción de acceder al menú del administrador, y la de entrar o registrarse

En la parte central de la pantalla tenemos la información del tema que suena, así como la opción de votar (solo mostramos que estará allí, la parte del controlador es quien se encargara de tramitar la acción) En uno de los laterales, tenemos el buscador de temas. De forma similar a como sucede con los votos la acción la llevara el controlador. El resultado no se mostrara en esta pantalla, sino en otra, que solo mantendrá, el menú de navegación respecto la pantalla que vemos

En el caso de la vista del administrador se ha usado un estilo sencillo y de acceso rápido a todas las funcionalidades

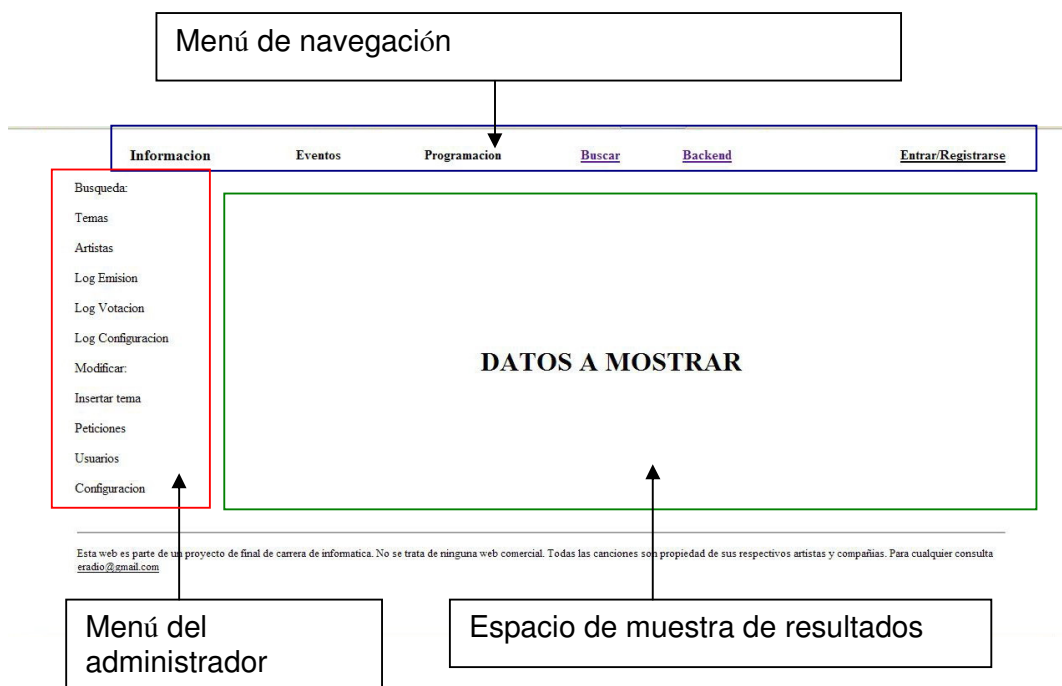


Figura 5: Vista de la interfaz del administrador

La vista del administrador mantiene la misma barra superior de funcionalidades que tiene la vista del usuario.

La parte central es la que difiere de la vista del usuario. Por un lado tenemos un menú lateral que se mantendrá siempre que estemos en la sección del administrador. En este menú estarán las opciones que podrá realizar el administrador. Se mantiene el menú de navegación para que el administrador pueda ejecutar los mismos procesos que cualquier usuario normal.

Se reserva una parte amplia de la pantalla, para que en ella aparezcan los resultados que genere el controlador de las peticiones del menú de administrador

2.5. Modificación en la Base de datos tras el primer diseño de la interfaz

Tras haber hecho el primer diseño de la interfaz, se revisará el diseño de la base de datos, para ver si pudiera existir algún caso que no se haya tenido en cuenta

En este caso, son dos tablas, de carácter funcional, los elementos que no se habían considerado. Estas son: La tabla de usuarios, y la tabla de Log de emisión

En la tabla de logemision, guardamos, tanto el tema, como el artista, así como el momento en el cual el tema fue emitido. El motivo de guardarlo en una tabla en vez de hacerlo en un fichero a parte, es la posibilidad de crear de manera más rápida cualquier tipo de consultas al respecto de la emisión de los temas, así como un sistema de referencia sobre el tema que está sonando en ese momento

La tabla de usuarios está compuesta, por el momento, por las columnas imprescindibles, que serán, nombre, password y dirección de mail, ésta última para poder contactar con el usuario

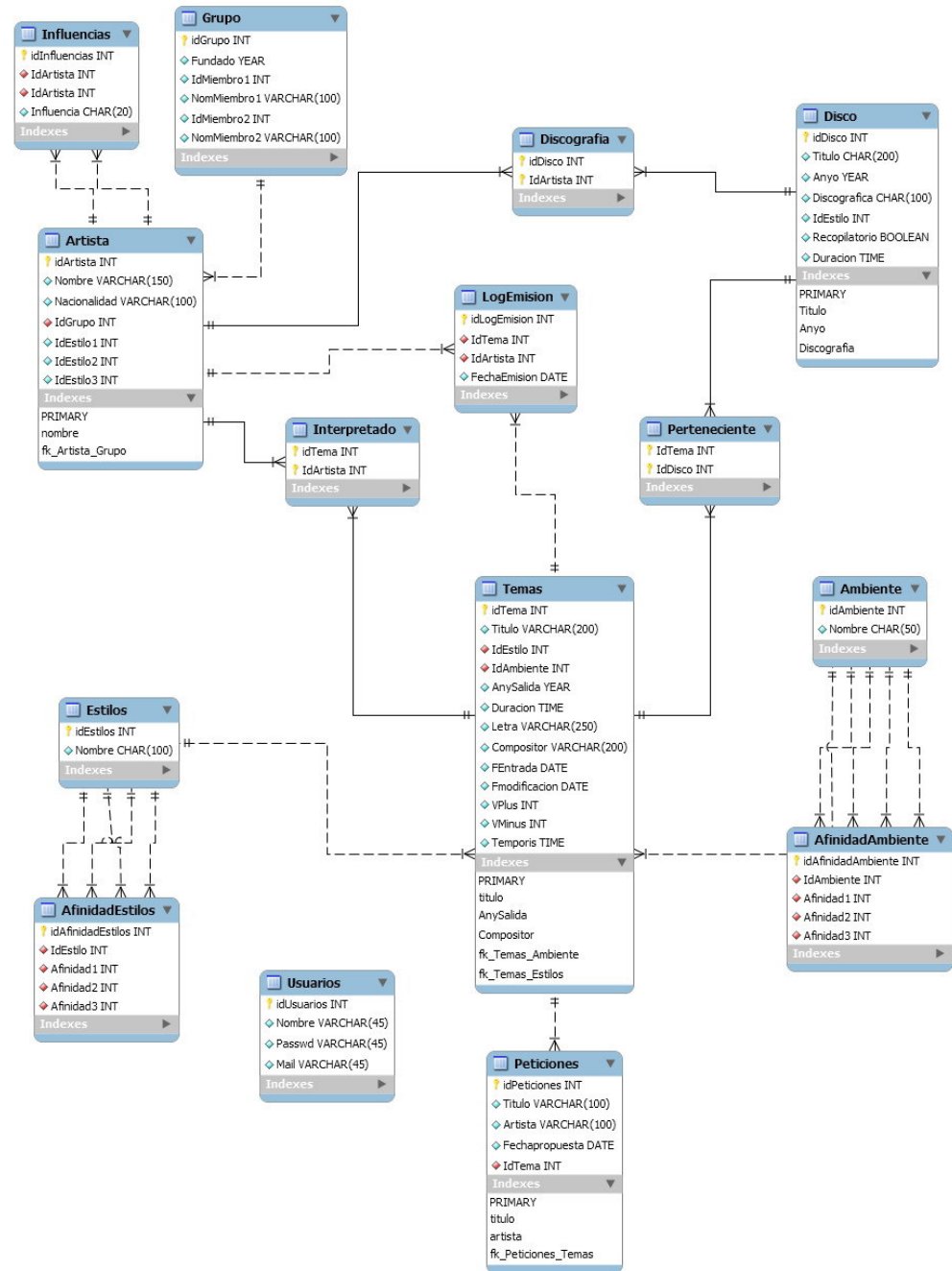


Figura 6: Base de Datos tras el análisis de vistas

3. Algoritmo de decisión

La emisora que estamos creando no es estática, sino que a través de los votos de la gente, así como de las propuestas, ira emitiendo unos temas o otros. La parte que realiza este proceso de selección, decidiendo que tema se ajusta mas a los gustos es el algoritmo de decisión

3.1. Sistema de representación

“Para resolver problemas de IA es necesario tener un cierto conocimiento del problema así como un conjunto de mecanismos para la resolución. En muchos de los problemas de IA la representación del conocimiento que se define puede simplificar mucho la resolución del problema” [Apuntes de Inteligencia Artificial I]

Nuestro primer paso en la decisión del algoritmo es, por lo tanto, buscar un sistema de representación de la información de los temas. Pero esto no es una tarea tan sencilla como pudiera parecer.

Uno de los principales problemas que existen a la hora de trabajar en el campo musical es la falta de un criterio único definido. Cada persona o grupo califica los temas y artistas de forma muy dispar, y no es infrecuente encontrar un mismo tema clasificado de mil y una maneras. Y si participan aficionados la clasificación es aun más caótica. En los foros musicales, es bastante común ver más de una disputa al respecto.

Es por eso, que para ese proyecto se decidió buscar una base (o guía de estilo) musical y basarse en ella. Con ello se evitarían, problemas de compatibilidad así como variación de criterios.

Después de buscar Webs especializadas en música, para este proyecto se ha escogido como guía musical la Web “Allmusic”. En esta página se

encuentra una abundante información, así como clasificación y relación en lo referente a artistas y temas.

Con una base musical de donde extraer la información que necesitamos, el siguiente paso es establecer las características que se puede extraer de nuestra guía.

Para realizar este estudio se selecciona una serie de temas candidatos y se intenta de extraer el máximo de características posibles. Tras el estudio, podemos ver que tenemos tres características: estilo, ambiente y relación entre grupos.

Una vez seleccionados estas características, se ha de crear un sistema de representación basado en las mismas. Para llevarlo a cabo, se creo una lista de temas y/o artista, que fuera lo suficientemente representativa del estilo de radio que deseábamos hacer, y se busco la información necesaria sobre los estilos y ambientes que estaban afectados para conseguir una base del sistema de representación.

No se realizo un estudio en el aspecto de relación de grupos, ya que para este aspecto solo se tendrá en cuenta la relación entre artistas que estén en la base de datos.

Del estudio de representación, se saco cerca de 125 estilos y 60 ambientes. Este valor, dado su tamaño y el valor marginal de alguno de los valores (por la cantidad de veces que salían), hizo que se crearan agrupaciones, para mejorar la interrelación de las características.

Para esta agrupación se utilizo los siguientes criterios: por un lado escoger aquellas características que estuvieran mas representadas, agrupar aquellas que tuvieran relación (en el caso de los estilos, se baso en la lista de subgéneros del rock, extraída de la propia pagina de Allmusic), y por ultimo que existiera alguna relación entre las características, de manera que si

representábamos la característica (ya sea estilo o ambiente) en un grafo, se pudiera acceder a cualquier nodo desde cualquier otro

De esta manera la lista de estilos resultantes queda de la siguiente manera:

- adult alternative pop/rock(1)
- adult contemporary(2)
- album rock(3)
- alternative pop/rock(4)
- arena rock(5)
- britpop(6)
- college rock(7)
- dance-rock(8)
- glam rock/art rock(9)
- grunge(10)
- hard rock(11)
- heavy metal(12)
- indie rock(13)
- metal-rock(14)
- new wave(15)
- pop(16)
- pop/rock(17)
- punk(18)
- rap-rock(19)
- rock&roll (20)
- singer/songwriter(21)
- pop jazz(22)

Para ver su grafo de relacion, mirar la figura 7

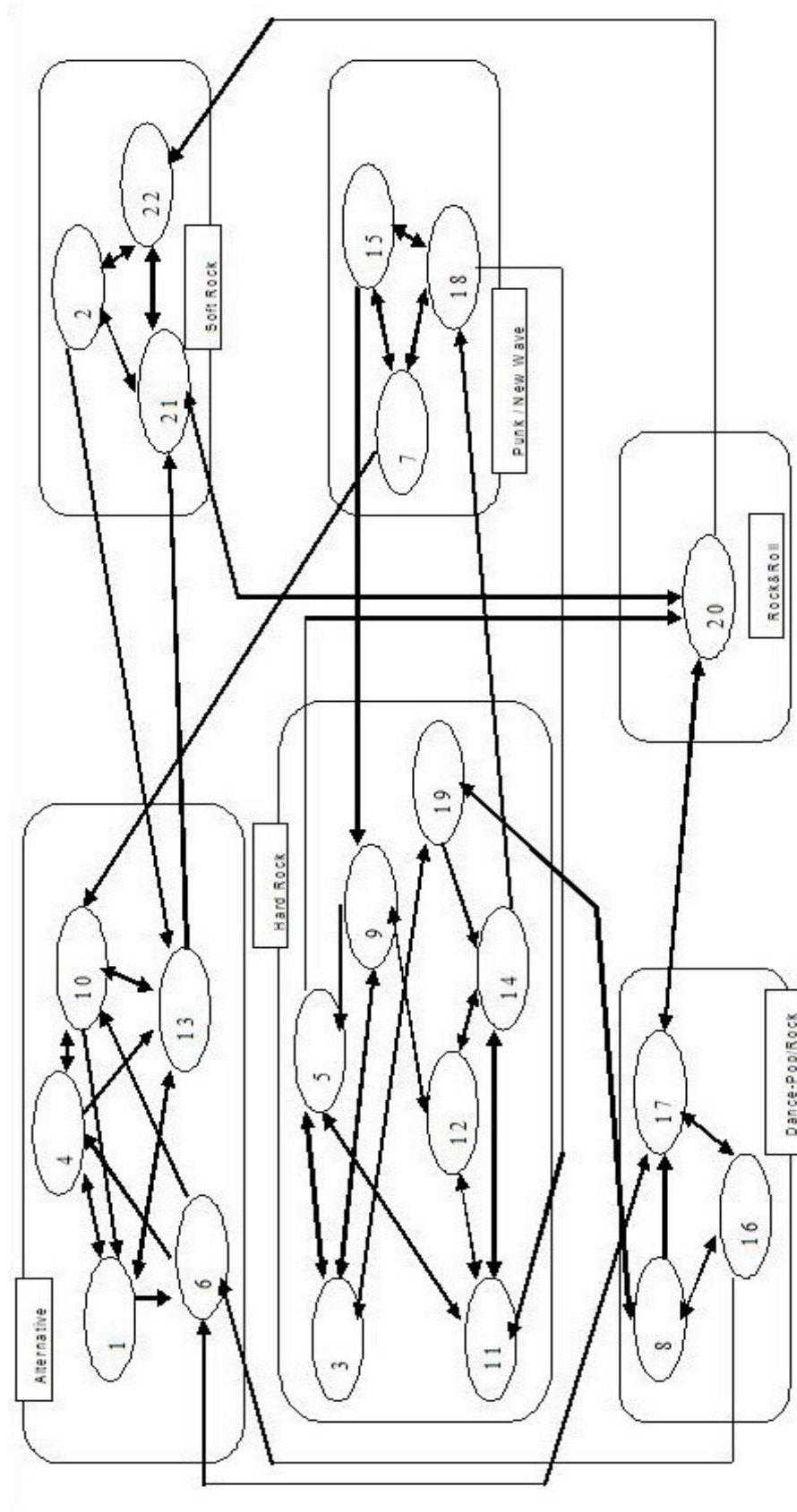


Figura 7: Grafo de relación de los estilos escogidos

Y la lista de ambientes de la siguiente:

- Optimista
- Apasionado
- Placentero
- Sensual
- Intenso / Enérgico
- Activo / Animado
- Austero
- Suave
- Relax
- Agresivo
- Agridulce
- Melancólico / Sentimental
- Triste

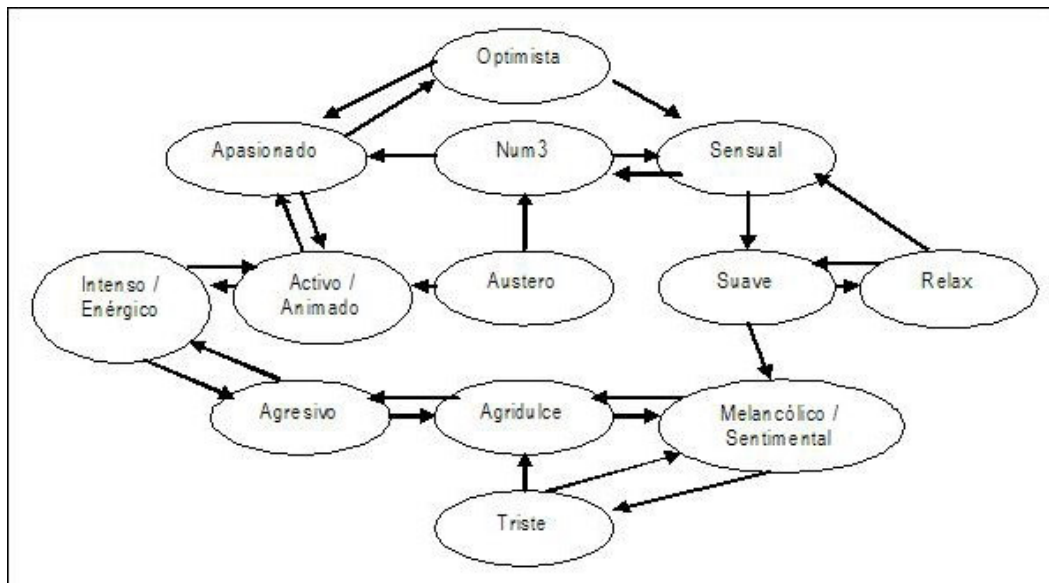


Figura 8: Grafo de relación de los ambientes escogidos

3.2. Estudio funcionalidad del Algoritmo

Una vez tenemos nuestro sistema de características, se analizan las restricciones y valores que tenemos para decidirnos por que algoritmo se ha de seleccionar. La primera restricción es que las tres características que tenemos, son independientes. Así se usara un algoritmo de "Búsqueda de satisfacción de restricciones"[1] por cada característica del nodo raíz (la canción que se esta emitiendo en este momento), se generaría un árbol con N nodos hijos (representando a las N canciones candidatas [2]) y de cada uno de estos nodos tendría que generar a su vez los N nodos otra vez, para poder evaluar otra características, y así sucesivamente.

- [1] Usamos como algoritmo de búsqueda, el algoritmo de "Búsqueda de satisfacción de restricciones", ya que se trata de buscar una solución, no una definida, y respetando unas determinadas restricciones, es decir, que tenga la menor distancia respecto al nodo que representa la canción que se esta emitiendo
- [2] Hablamos de N canciones candidatas, ya que el administrador podrá especificar en la configuración de la emisora el número de canciones candidatas a ser evaluadas en cada interacción

En una situación así, y con la cantidad de características que tenemos, la solución mas adecuada es la utilización de un algoritmo basado en la selección de espacio de características.

Cada uno de nuestros temas, estará representado en un espacio de características de tres ejes (estilo, ambiente y relación de grupos), y el sistema para calcular cual será el nuevo tema seleccionado realizara el calculo de la distancia entre las características del tema candidato y las características del tema que esta sonando.

Pero este sistema tiene un problema implícito. ¿Que pasara cuando solo un usuario vote un tema? La decisión de una única persona puede modificar

todo el estilo y ambiente de la emisora. En los casos de las emisoras privadas, un voto negativo genera un cambio bastante drástico en las características de los temas que están sonando. ¿Como afectara esto a nuestra emisora?

La solución para este posible conflicto es la creación del estilo de emisora. En la configuración de la emisora se definirá un estilo y ambiente para la emisora. No se seleccionara ningún grupo, esta característica se seguirá sacándose del tema que esta sonando, ya que no esta acotado en numero como es el caso de ambiente y estilo, además al no estar definido permite que si se selecciona un tema aleatorio, si fuera necesario, se utilizara las características definidas de estilo y ambiente. Si también definiéramos el grupo, las opciones de emisión de un tema aleatoria se reducirán drásticamente (o podrían ser nulas, si todos los temas de ese artista están bloqueados).

Esta opción a la vez permite que el administrador pueda modificar las características de la configuración de la emisión antes de iniciar la misma.

Aun así, hay que considerar un tema. Se ha estado hablando de que se trata de una emisión dinámica, ¿no implica este cambio un contrasentido con lo que se ha estado exponiendo hasta ahora? Si dejáramos que este estilo y ambiente se mantuviera siempre, así seria, pero no es el caso.

Cada vez que suena un tema, y los usuarios lo voten, da igual si se trata de votos positivos o negativos, este conjunto de votos totales es evaluado. Si la mayoría de usuarios realizan un voto positivo, el estilo y ambiente del tema que ha sonado recibirán un aumento de sus votos (así como los estilos y ambientes asociados) y una disminución si la mayoría de votos es negativa. Cada cierto tiempo (a definir por el administrador en numero de canciones), se evaluara esta tabla de votos de estilos y ambientes. El estilo y ambiente mas votados, se convertirán, en los nuevos estilo y ambiente de la emisora. Si el número de temas definido para calcular la modificación de la configuración es bajo, la radio modificara sus características de manera más rápida a partir de los votos de los usuarios, y un número de temas mas alto permitirá evitar que un solo usuario con sus votos modifique el estilo de la emisora. El valor inicial

marcado es de 5 temas, pero es decisión del administrador, en función de la velocidad de adaptación a los votos, el que marcara el valor final.

3.2.1. Sistema de coordenadas

Así tendremos un espacio de características de tres ejes (estilo, ambiente y relación entre grupos) donde hemos de calcular la distancia de dos muestras entre si (la muestra definida por la emisora y cada uno de los temas candidatos)

Para facilitar el calculo, se considera que el punto (0,0,0), representara el estilo de la emisora + el grupo q esta sonando, y que los tres ejes tendrán representados los nodos en relación con la característica o afinidad de la misma. De esta manera cuando un nodo este más cerca del punto (0,0,0) más posibilidades tendrá de ser seleccionado

3.2.2. Peso de las características:

Otro aspecto que debemos definir es el peso de cada una de las características. A tal efecto se han tenido en cuenta dos consideraciones: uso que realizamos de las mismas, y peso que se le asigna en otros servicios similares al nuestro

En relación al uso que le damos, tal y como se ha comentado antes, en la asignación de la configuración de la emisora, solo se tiene en cuenta el ambiente y el estilo, mientras que la relación entre grupos queda relegada a la selección del tema que esta sonando en este momento.

Así, aunque la información del grupo, es más directa sobre el tema, es menos interesante para la configuración de la emisora, ya que buscamos las características que mantengan un estilo de emisora, no las características mas directas, ya que estas pueden verse afectadas, si un solo usuario vota las canciones

En relación al uso de servicios similares, se ha enviado consultas a varios servicios, principalmente las emisoras personalizadas, para saber con que características trabajan. La mayoría de las emisoras que han contestado, con la excepción de Last.fm se basan en características del tema, sin considerar las relaciones existentes en (como caso a destacar, hablaríamos de Pandora, que llegan a usar mas de 400 características de los temas, para decidir cual es el siguiente tema a emitir, sin que sea ninguno de ellos la relación entre artistas)

Una vez considerados estos factores, el peso especificado de las características quedara definido de la siguiente manera: estilo y ambiente tiene un peso muy importante, por lo que se les asignara un 45% del total a cada uno, mientras que el peso de la relación de grupo es marginal, y servirá básicamente para acabar de seleccionar entre dos temas, con características muy similares. Por tanto solo tendrá un peso de un 10%.

3.2.3. Factor de novedad

Otro factor a tener en cuenta a la hora de decidir cual es el siguiente tema a sonar es el factor de novedad. Nos interesara que un tema que acaba de entrar en nuestra emisora tenga más posibilidades de emitirse que otro que lleva mas tiempo emitiéndose. Para poder realizar este control, se crea un script que se ejecutara semanalmente (o cuando lo decida el administrador). El script mirara la fecha de entrada del tema, y comparándola respecto la fecha actual, asignara un valor de novedad, de valor decreciente en tanto que haya pasado mas tiempo, hasta llegar al valor de neutro una vez el tema haya pasado un mes en la emisora.

Para representar este valor de novedad, se asignara como valor por defecto, cuando un nuevo tema sea introducido de 8, siendo dividido por dos a cada semana que pase, hasta llegar a un valor de 1, cuando se considera que el tema ya no es una novedad.

3.2.4. Calculo de la distancia

Una vez considerados todos estos conceptos, la distancia de un tema, respecto a la configuración de quedara de la siguiente manera

$$1/F(\sum (X)(P))$$

Donde

- F corresponde al factor de novedad
- X al valor de la característica
- P al peso de la característica

3.2.5. Otras consideraciones a tener en cuenta

Votos asociados

¿Como recibe los votos un tema? Existen dos posibilidades, por un lado cuando un usuario vota el tema mientras este sonando; y por otro lado los votos asociados. Un voto asociado es aquel voto que recibe un tema si su estilo o característica son iguales a las características del tema que esta sonando.

Así si un tema recibe mas votos positivos que negativos, se considerara que los temas, con esas características son los que quieren los usuarios, por tanto, todos los temas, con alguna de esas características recibirán votos asociados. De igual manera si los votos que recibe un tema son mas negativos que positivos. Si los votos son iguales, o no se ha realizado ningún voto, ningún tema recibirá ningún voto asociado

Bloqueo de canciones

Por ultimo, a la hora de considerar el algoritmo, se ha de tener en cuenta, que se debe realizar un bloqueo de la canción que acaba de sonar, para evitar que vuelva a sonar en un cierto tiempo. Es por eso que un tema que acaba de sonar se le bloqueara durante un periodo de tiempo (1 hora aproximadamente), durante la cual, no podrá ni recibir votos (ya sean directos o asociados), ni ser candidato a ser seleccionado. A la vez, a la hora de desbloquearlo, se le actualizara los valores de votos que tuvieran, tanto los positivos como los negativos, en relación a la media de la emisora.

3.3. Algoritmo

Mientras la radio este activa

1. Buscar información de las características de la emisora (estilo, ambiente, calculo configuración, limited, tiempo de bloqueo)
2. Si no hay ninguna información de la emisora ->
 - 2.A.I. Error, la radio no puede funcionar sin una configuración definida
3. Si hay información de la emisora
 - 3.A. Buscar la información del tema que se emite
 - 3.B. Si no hay información
 - 3.B.I. Buscar un tema que cumpla las características de la emisora
(estamos arrancando el sistema)
 - 3.B.II. Cargar tema
 - 3.B.III. Salir del algoritmo
 - 3.B.IV. Si no hay ningún tema que cumpla alguna de las
características de la emisora, se emitirá el primer tema que exista
4. Si no ha habido ningún error:
5. Buscar características asociadas a características (estilo) de la emisora
6. Buscar características asociadas a características (ambiente) de la emisora
7. Buscar artista del tema que suena
8. Buscar los N (N=limited) temas no bloqueados, ordenados por (vplus-vminus)
9. Por cada tema seleccionado
 - 9.A. Buscar relación característica (estilo) tema candidato con característica configuración
 - 9.B. Buscar relación característica (ambiente) tema candidato con característica configuración
 - 9.C. Buscar relación de grupo tema candidato con tema que suena
 - 9.D. Buscar factor de novedad del tema candidato

9.E. Calcular distancia del tema respecto características de la emisora:

$$\text{Distancia} = 1/(\text{f. novedad}) * (\sum (\text{Relación característica}) * (\text{Peso característica}))$$

9.F. Mirar si es el tema candidato tiene una distancia menor que la distancia actual

10. Bloquear tema seleccionado

11. Calcular $VR = v_{\text{plus}}/v_{\text{minus}}$ (relación de votos positivos y negativos) del tema actual

12. Si ($VR > 1$), el tema gusta

12.A. Se suma un voto positivo a todos los temas no bloqueados por cada característica común (estilo, ambiente) que tengan con el tema que ha sonado

12.B. Se calcula nuevos valores de las características (estilo, ambiente)

13. Si ($VR = 1$) indiferente, no se actualiza nada

14. Si ($VR < 1$) el tema no gusta

14.A. Se suma un voto negativo a todos los temas no bloqueados por cada característica común (estilo, ambiente) que tengan con el tema que ha sonado

14.B. Se calcula nuevos valores de las características (estilo, ambiente)

15. Calcular si toca recalcular de la configuración del sistema (calculoconfiguracion)

16. Calcular valor medio para votos positivos y negativos para la actualización

17. Desbloquear temas bloqueados que hayan pasado el tiempo establecido (tiempo de bloqueo) con los valores de v_{minus} y v_{plus} iguales a la media de los votos

18. Cargar nuevo tema en el log de emisión

19. Poner nuevo en la lista de reproducción

3.4. Modificación en la Base de datos tras la selección del algoritmo de decisión

Tras la selección del algoritmo de decisión, se comprueba la base de datos. Existen dos factores que pueden afectar a la base de datos. Uno es de carácter funcional, con la creación ya sea de tablas o bien columnas, para poder ejecutar correctamente el algoritmo. Y el otro factor, es la información “redundante” de la tabla de datos, que se puede eliminar de la base de datos.

El motivo que tendremos para eliminar esta información, es sobretodo, por el peso específico que esta información comporta. A la hora de seleccionar los pesos de las características, la menos importante es la relación entre grupos, siendo importante las características del tema, no el grupo que lo interpreta.

Es por eso, que se limita la información relacionada con grupos. Datos como los miembros de un grupo, no necesitan tener una tabla a parte, ya que si solo se utilizan como información, pueden existir solo como un campo en la tabla de artistas. Por otro lado, y dentro de esta política, los datos relacionados con disco, también son poco significativos.

A esta decisión también ayuda, que la información relativa a inclusión de un tema en un disco, no es equitativa para todos los temas que se han buscado en Allmusic (siendo bastante común que ciertos temas, carezcan de la misma), se ha decidido, reducir esta información, al disco donde aparecía originalmente el tema

Así, los pasos, para reducir la información poco significativa han sido:

- Eliminar la tabla “grupo”
- Creación del campo “componentesgrupo” en la tabla artista
- Eliminar los campos de “estilos” en la tabla “artista”
- Eliminar la tabla “discografía”

- Eliminar la tabla “perteneiente”
- Eliminar la tabla “disco”
- Creación de los campos de información necesaria sobre el disco, en la tabla temas: “nombredisco”, “discográfica”

En el caso de las modificaciones realizadas para la funcionalidad, estos son los pasos que se han realizado:

Para controlar los votos asociados a las características, se crea las tablas votambiente y votosestilo. Con estas tablas, controlaremos los gustos de los usuarios en lo referente a estas dos características, y a través de ellos, se podrán escoger que estilo y que ambiente es el que seleccionaremos como definido en la emisora

La selección de estas características, así como de otras, estarán definidas en la tabla “configuracionsistema” que también se creara

Para llevar un control tanto de las diferentes configuraciones que se sucederán, así como los votos que se emitirán, se crea dos tablas, una “logvotacion” llevara el control del voto (positivo o negativo) , el tema que ha recibido el voto, así como el usuario que lo ha hecho. Y por otro lado “logconfiguracion”, que guardara la información de las distintas configuraciones que se irán generando

En lo referente al control del bloqueo del tema, se crea un nuevo campo en la tabla temas, “bloqueoemision” de tipo booleano, que controlara si el tema esta en espera, y por otro lado temporis que nos diría en que momento fue bloqueado. A través de este campo veremos si ha pasado el tiempo necesario para desbloquear el tema.

Por otro lado se crear el campo “bloqueotema”. Este campo servirá para eliminar lógicamente un tema. Un tema nunca se eliminara de manera física de la base de datos, ya que en algún momento este tema ha sonado, y por ese

motivo guardaremos la información, por si tuviéramos que consultarlo en un futuro.

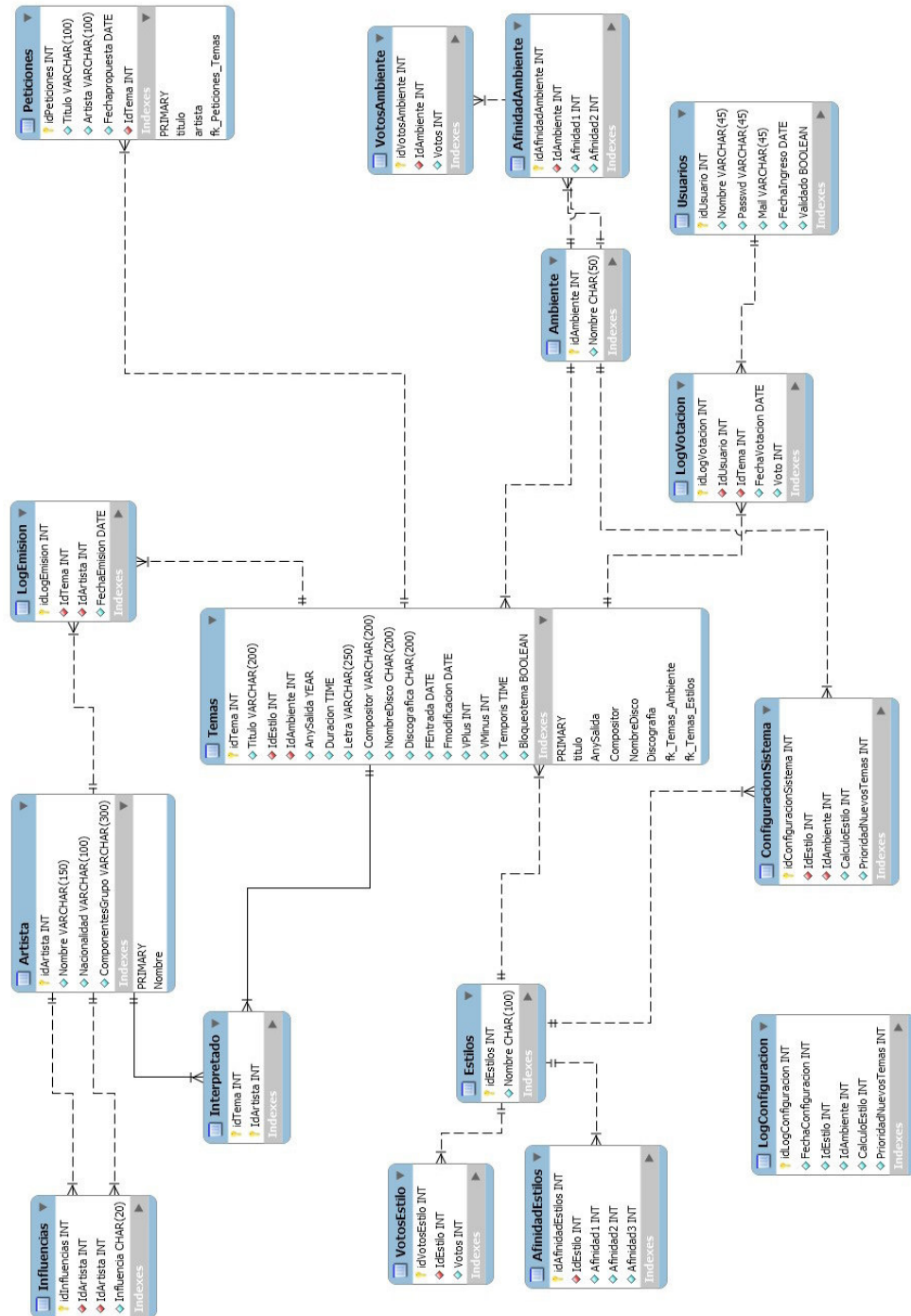


Figura 9: Base de Datos tras la definición del algoritmo de decisión

4. Pruebas realizadas

Se han realizado dos tipos de pruebas. Por un lado las pruebas de las funcionalidades de control, cuyo objetivo era mostrar información, o interactuar con ella (búsqueda de temas, visualización de logs, votación de temas) y por otro lado las pruebas de la ejecución del algoritmo.

Para el primer tipo de pruebas, se han creado pequeñas páginas de chequeo, cada una con el objetivo de realizar cada una de las tareas encomendadas.

Y por otro lado las pruebas referentes al algoritmo de decisión. Se han realizado varias pruebas, tanto para comprobar los posibles errores de configuración que existieran, así como arrancando el sistema y por tanto teniendo que escoger un tema al azar, o con una serie de temas ya emitidos.

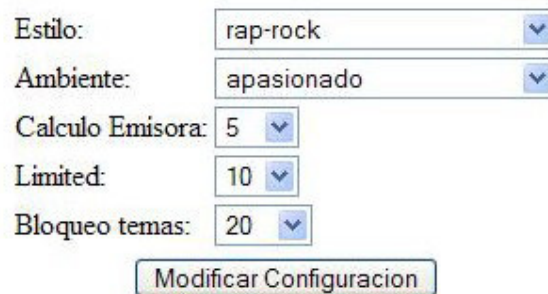
Una de las modificaciones que se ha realizado durante las pruebas, es el sistema de bloqueo de los temas. Con tal de proporcionar al administrador una nueva opción de configuración de la emisora, ya no se usa un sistema basado en el tiempo para definir el tiempo que un tema estará bloqueado tras emitirse

El nuevo sistema, permitirá al administrador definir cuantos temas estará bloqueado tras ser emitido.

Tal y como hacemos para decidir si toca recalcular o no las características, usaremos el idlogemision para saber cuantos temas se ha emitido, y a partir de él definir si se ha de recalcular las características, así hasta que tema se ha de esperar un tema bloqueado

Ahora veremos un ejemplo de la ejecución del algoritmo

El primer paso en una nueva emisión, consistirá en configurar la emisora (o bien dejar la configuración ya establecida)



Estilo: rap-rock

Ambiente: apasionado

Calculo Emisora: 5

Limited: 10

Bloqueo temas: 20

Modificar Configuración

Figura 10: Configuración de la emisora

Con la emisora configura, en el momento que el suceso de nuevo tema se ejecute, buscaremos la información del tema que acaba de sonar, la configuración de la emisora, así como las características directas e indirectas, tanto de la emisora como del tema que acaba de sonar.

Se seleccionara una cantidad de temas determinado, los cuales se comprobara las características respecto las de la emisora, con lo cual nos dará una distancia

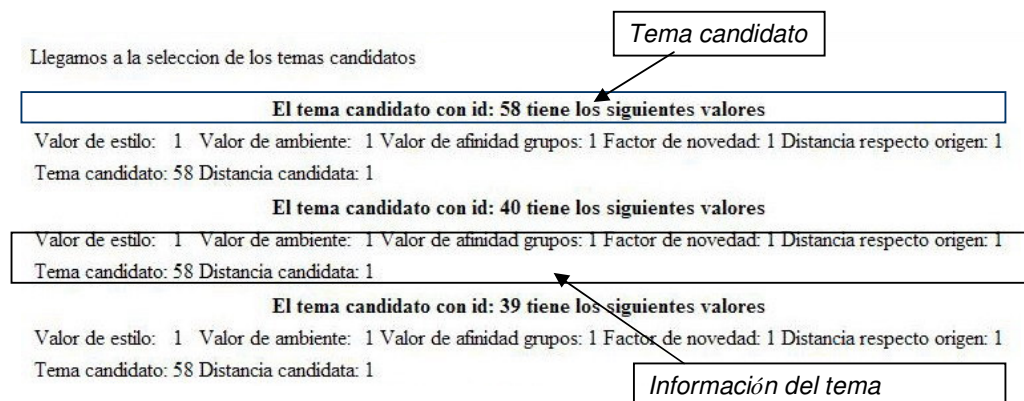


Figura 11: Lista de temas candidatos

El nuevo tema, será aquel que nos de una distancia menor (Para un ejemplo completo, paso por paso, consultar el anexo)

4.1. Modificaciones en la base de datos tras las pruebas

Durante el proceso de las pruebas realizadas, se han comprobado una serie de errores, o mejoras realizables a la base de datos. Estas mejoras consisten en:

Se ha eliminado de la tabla de logemision, la columna artista. Ya que un tema puede ser interpretado por varios artista, no tiene sentido guardar esta información, ya q a través del tema, sabremos que artista o artista lo interpretaron

La tabla de “votosestilo” y “votosambiente” no aportan tanto como para necesitar una tabla propia. Para no tener tantas tablas, se ha pasado la información de los votos de estilo y ambiente a las propias tablas de estilo y ambiente. Con tal de evitar problemas de bloqueos, ambas tablas pasan de ser tablas MyIsam a tablas InnoDB. También, en este aspecto de eliminar columnas redundantes, se modifica la columna de idafinidadestilo y idafinidadambiente, y se usa directamente idestilo, y idambiente. Aunque se podrían haber integrado en la tabla de estilo y ambiente, se han mantenido separadas para permitir mas relaciones futuras de los diferentes estilos o ambientes

Se ha agregado la columna “link”, a la tabla temas. En este campo guardaremos la información de donde se encuentra el tema, para que pueda ser reproducido por el programa de emisión que decidamos.

Para poder realizar la eliminación lógica de los temas, se ha añadido la columna eliminado a la tabla temas, de tipo booleano. De esta manera podremos marcar que temas, han sido eliminados lógicamente y cuales no

Otra modificación que se ha realizado ha afectado a la tabla de peticiones. Se ha creado una nueva columna idusuario, donde se almacenara la id del usuario que proponga el tema. Esta nueva información servirá para informar al usuario que ha propuesto el tema cuando el tema ha sido introducido en la emisora, o para pedir mas información sobre el tema candidato.

Por ultimo, para poder permitir la modificación del funcionamiento del sistema de bloqueo de temas, se han realizado los siguientes cambios:

En la tabla de configuracionsistema y logconfiguracion, se han creado la columna Bloqueotemas, donde se definirá la cantidad de temas que un tema estará bloqueado. Para poder controlar este valor, se ha modificado la columna temporis de tipo time a tipo INT

Tras estos cambios la base de datos, quedara de la siguiente manera

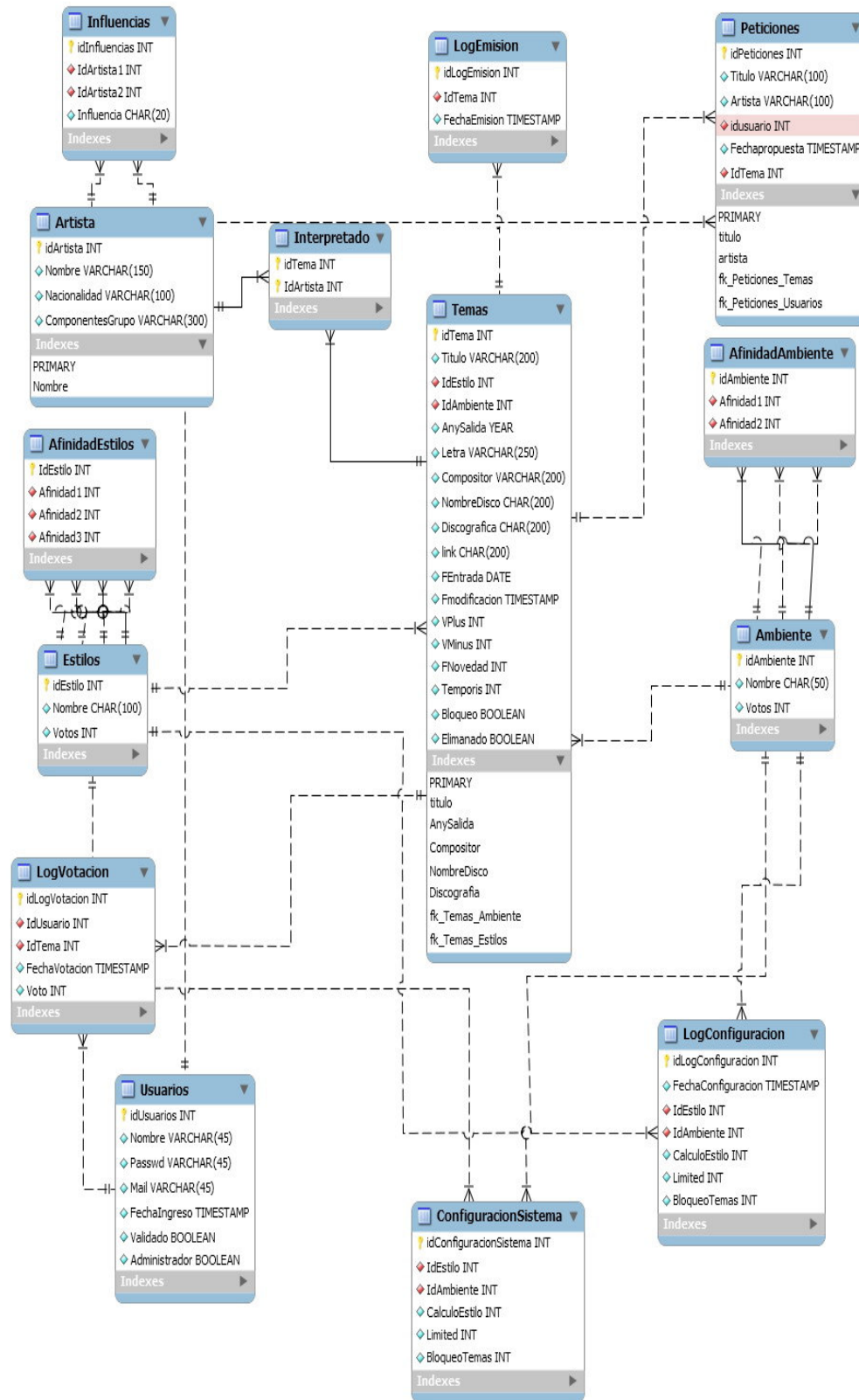


Figura 12: Base de Datos final

5. Conclusiones

5.1. Objetivos conseguidos y Temas pendientes

No se han podido conseguir todos los objetivos fijados en un primer momento. Si que se ha realizado un sistema de decisión que teniendo en cuenta los gustos de la gente es capaz de seleccionar el tema mas adecuado a sonar a partir de unos temas candidatos, así como un interfaz sencilla para la interacción de los usuarios.

Lo que no se ha podido conseguir es la integración de la herramienta shoutcast en el interfaz de la emisora, con lo que, el sistema escoge el tema, pero no se puede realizar la emisión.

El principal tema pendiente es la conexión del interfaz con el sistema shoutcast, para que no solo se pueda decidir el tema a sonar, sino que este se pueda emitir

Otro tema interesante a mejorar es el aspecto de la interfaz. Se ha seleccionado una interfaz sencilla, y de fácil manejo, pero si se desea que esta emisora fuera funcional, tendría que mejorarse considerablemente

5.2. Futuras mejoras

- Mejora del espacio de características:

Uno de los principales aspectos que se puede mejorar es el referente al espacio de características. En el proyecto solo se ha trabajado con tres características, pero es ampliable a tantas como se crea necesario, en especial aquellas que traten de nuevas características del tema. El principal impedimento es encontrar la fuente de información fiable para realizar esta información.

Como ejemplo, ya comentado antes, está el caso Pandora que posee más de 400 atributos. Pero este es un caso hipotético, y que no se podría realizar sin ayuda de especialistas en música, tal y como ellos mismos comentan:

“Pandora has well over 400 attributes. They are very precise things such as which beats the kick drum is used, What pattern is played on the hi-hat, how much vibrato the lead singer uses, what chord changes, etc. Most of these things are things a university trained musician can be trained to hear.”

- Mejoras en las opciones del administrador:

Incluir opciones de búsquedas más avanzadas de los diferentes logs (para poder sacar estadísticas más correctas), como de información de la emisora; así como permitir más aspectos configurables de la emisora (como podría ser los pesos de las características)

- Intersección de cuñas publicitarias:

Uno de los problemas del algoritmo es el retardo que produce entre tema y tema. Una manera de solucionar este problema a través de las cuñas publicitarias.

La idea es que al acabar de sonar un tema, no se lanzaría el siguiente tema, sino que se lanzaría una pequeña cuña publicitaria, ya sea de la emisora o de un anunciante. Durante la cuña, el algoritmo calcularía cual es el siguiente tema a sonar, sin que eso reporte un corte de la emisión de la emisora.

De esta manera no solo evitaríamos los molestos segundos de espera entre temas, sino que aseguraríamos una fuente de ingresos extras para la emisora.

- Votos de temas buscados:

Permitir que un usuario pueda votar por un tema, al buscarlo. Esta funcionalidad solo sería factible, si comportara a la vez, un control de manera que un usuario solo pudiera votar un número limitado de veces a un tema, durante la emisión de un tema. Un posible sistema sería guardar en el log de

votación, no solo el tema y el voto, sino que además, guardar el log de emisión, para controlar las veces que vota durante la emisión de un tema

Otra característica interesante, sería ampliar la información que se guardar de un usuario, para poder notificarle eventos, actividades o temas nuevos que podrían interesarle a partir de su ínter actuación con la emisora.

6. Herramientas utilizadas

Para la realización del proyecto se han utilizado las siguientes herramientas:

- Para la Base de Datos:

Para la realización de la base de datos se ha usado MySQL “un sistema de gestión de base de datos relacional, multihilo y multiusuario”

Se han usado el Mysql Workbench para el diseño y creación de la base de datos y por otro lado el paquete de MySQL GUI Tools (en concreto el MySQL Administrador para el control de la Base de Datos, y por otro lado MySQL Query Browser para las pruebas de la base de datos)

- Para la interfaz de la radio:

Para la realización de la página se ha usado el lenguaje de programación PHP sobre un servidor apache conectado a la base de datos MySQL

El entorno de desarrollo que se ha usado es Eclipse+PDT, el cual permite trabajar en todas las capas que componen la pagina Web

- Para la reproducción se ha usado el programa Winamp junto con el plug-in shoutcast

Las ventajas de la utilización de todos estos componentes, es que todos ellos son software libre y multiplataforma, de tal manera que pueden ser usados tanto por plataformas Windows como Linux, a parte de implicar un coste nulo, cuestión importante a la hora de plantear la viabilidad de este proyecto

7. Bibliografía

Bibliografía

Programas usados	
Mysql	http://www.mysql.com/
Mysql workbench	http://mysql.com/products/workbench/
Mysql GUI Tools	http://mysql.com/products/tools/
Php	http://www.php.net/
Eclipse	http://www.eclipse.org/
Winamp	http://www.winamp.com/
Shoutcast	http://www.shoutcast.com/

Programas de radios online consultados	
Lastfm	http://www.lastfm.es/
Pandora	http://www.pandora.com
Rockola	http://rockola.fm/
Mucicoverly	http://musicoverly.com
Finetune	http://www.finetune.com/

- Información Adicional

- Apuntes de Inteligencia Artificial I
Maria Vandrell
- Apuntes de Inteligencia Artificial II
Jordi Vitria
- Apuntes de Base de Datos I
Enric Marti
- Apuntes de Base de Datos II
Xavier Binefa
- Foros del web
<http://www.forosdelweb.com>
- Allmusic
<http://www.allmusic.com/>

Anexos

Modelo Físico de la Base de Datos

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='TRADITIONAL';

CREATE SCHEMA IF NOT EXISTS `eradio` DEFAULT CHARACTER SET latin1
COLLATE latin1_general_ci ;
USE `eradio`;

-- -----
-- Table `eradio`.`Artista`
-- -----
CREATE TABLE IF NOT EXISTS `eradio`.`Artista` (
  `idArtista` INT NOT NULL AUTO_INCREMENT ,
  `Nombre` VARCHAR(150) NOT NULL ,
  `Nacionalidad` VARCHAR(100) NULL ,
  `ComponentesGrupo` VARCHAR(300) NULL ,
  PRIMARY KEY (`idArtista`) ,
  INDEX Nombre (`Nombre` ASC) )
ENGINE = MyISAM;

-- -----
-- Table `eradio`.`Temas`
-- -----
CREATE TABLE IF NOT EXISTS `eradio`.`Temas` (
  `idTema` INT NOT NULL AUTO_INCREMENT ,
  `Titulo` VARCHAR(200) NOT NULL ,
  `IdEstilo` INT NOT NULL ,
  `IdAmbiente` INT NOT NULL ,
  `AnySalida` YEAR NULL ,
  `Letra` VARCHAR(250) NULL ,
  `Compositor` VARCHAR(200) NULL ,
  `NombreDisco` CHAR(200) NULL ,
  `Discografica` CHAR(200) NULL ,
  `link` CHAR(200) NULL ,
  `FEntrada` DATE NULL ,
  `Fmodificacion` TIMESTAMP NULL ,
  `VPlus` INT NOT NULL DEFAULT 0 ,
  `VMinus` INT NOT NULL DEFAULT 0 ,
  `FNovedad` INT NOT NULL DEFAULT 8 ,
  `Temporis` INT NOT NULL ,
  `Bloqueo` BOOLEAN NOT NULL DEFAULT false ,
  `Eliminado` BOOLEAN NOT NULL DEFAULT false ,
  PRIMARY KEY (`idTema`) ,
  INDEX titulo (`Titulo` ASC) ,
  INDEX AnySalida (`AnySalida` ASC) ,
  INDEX Compositor (`Compositor` ASC) ,
  INDEX NombreDisco (`NombreDisco` ASC) ,
  INDEX Discografia (`Discografica` ASC) )
ENGINE = InnoDB;

-- -----
-- Table `eradio`.`Estilos`
-- -----
```

```
CREATE TABLE IF NOT EXISTS `eradio`.`Estilos` (  
  `idEstilo` INT NOT NULL AUTO_INCREMENT ,  
  `Nombre` CHAR(100) NOT NULL ,  
  `Votos` INT NOT NULL DEFAULT 100 ,  
  PRIMARY KEY (`idEstilo`) )  
ENGINE = InnoDB;  
  
-----  
-- Table `eradio`.`AfinidadEstilos`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`AfinidadEstilos` (  
  `IdEstilo` INT NOT NULL AUTO_INCREMENT ,  
  `Afinidad1` INT NOT NULL ,  
  `Afinidad2` INT NULL ,  
  `Afinidad3` INT NULL ,  
  PRIMARY KEY (`IdEstilo`) )  
ENGINE = MyISAM;  
  
-----  
-- Table `eradio`.`Influencias`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`Influencias` (  
  `idInfluencias` INT NOT NULL AUTO_INCREMENT ,  
  `IdArtista1` INT NOT NULL ,  
  `IdArtista2` INT NOT NULL ,  
  `Influencia` CHAR(20) NULL ,  
  PRIMARY KEY (`idInfluencias`) )  
ENGINE = MyISAM;  
  
-----  
-- Table `eradio`.`Interpretado`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`Interpretado` (  
  `idTema` INT NOT NULL ,  
  `IdArtista` INT NOT NULL ,  
  PRIMARY KEY (`idTema`, `IdArtista`) )  
ENGINE = MyISAM;  
  
-----  
-- Table `eradio`.`Ambiente`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`Ambiente` (  
  `idAmbiente` INT NOT NULL AUTO_INCREMENT ,  
  `Nombre` CHAR(50) NOT NULL ,  
  `Votos` INT NOT NULL DEFAULT 100 ,  
  PRIMARY KEY (`idAmbiente`) )  
ENGINE = InnoDB;  
  
-----  
-- Table `eradio`.`AfinidadAmbiente`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`AfinidadAmbiente` (  
  `idAmbiente` INT NOT NULL AUTO_INCREMENT ,  
  `Afinidad1` INT NOT NULL ,  
  `Afinidad2` INT NULL ,  
  PRIMARY KEY (`idAmbiente`) )
```

ENGINE = MyISAM;

```
-----  
-- Table `eradio`.`Peticiones`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`Peticiones` (  
  `idPeticiones` INT NOT NULL AUTO_INCREMENT ,  
  `Titulo` VARCHAR(100) NULL DEFAULT '' ,  
  `Artista` VARCHAR(100) NULL DEFAULT '' ,  
  `IdUsuario` INT NULL ,  
  `Fechapropuesta` TIMESTAMP NULL ,  
  `IdTema` INT NULL ,  
  PRIMARY KEY (`idPeticiones`) ,  
  INDEX titulo (`Titulo` ASC) ,  
  INDEX artista (`Artista` ASC) )  
ENGINE = MyISAM;
```

```
-----  
-- Table `eradio`.`LogEmision`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`LogEmision` (  
  `idLogEmision` INT NOT NULL AUTO_INCREMENT ,  
  `IdTema` INT NOT NULL ,  
  `FechaEmision` TIMESTAMP NOT NULL ,  
  PRIMARY KEY (`idLogEmision`) ,  
  INDEX FechaEmision (`FechaEmision` ASC) )  
ENGINE = Falcon;
```

```
-----  
-- Table `eradio`.`Usuarios`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`Usuarios` (  
  `idUsuarios` INT NOT NULL AUTO_INCREMENT ,  
  `Nombre` VARCHAR(45) NOT NULL ,  
  `Passwd` VARCHAR(45) NOT NULL ,  
  `Mail` VARCHAR(45) NULL ,  
  `FechaIngreso` TIMESTAMP NULL ,  
  `Validado` BOOLEAN NOT NULL DEFAULT false ,  
  `Administrador` BOOLEAN NULL ,  
  PRIMARY KEY (`idUsuarios`) )  
ENGINE = InnoDB;
```

```
-----  
-- Table `eradio`.`LogVotacion`  
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`LogVotacion` (  
  `idLogVotacion` INT NOT NULL AUTO_INCREMENT ,  
  `IdUsuario` INT NOT NULL ,  
  `IdTema` INT NOT NULL ,  
  `FechaVotacion` TIMESTAMP NOT NULL ,  
  `Voto` INT NOT NULL ,  
  PRIMARY KEY (`idLogVotacion`) )  
ENGINE = InnoDB;
```

```
-----  
-- Table `eradio`.`ConfiguracionSistema`  
-----
```

```
-----  
CREATE TABLE IF NOT EXISTS `eradio`.`ConfiguracionSistema` (  
  `idConfiguracionSistema` INT NOT NULL AUTO_INCREMENT ,  
  `IdEstilo` INT NOT NULL ,  
  `IdAmbiente` INT NOT NULL ,  
  `CalculoEstilo` INT NOT NULL ,  
  `Limited` INT NOT NULL ,  
  `BloqueoTemas` INT NOT NULL ,  
  PRIMARY KEY (`idConfiguracionSistema`) )  
ENGINE = InnoDB;
```

```
-----  
-- Table `eradio`.`LogConfiguracion`  
-----
```

```
CREATE TABLE IF NOT EXISTS `eradio`.`LogConfiguracion` (  
  `idLogConfiguracion` INT NOT NULL AUTO_INCREMENT ,  
  `FechaConfiguracion` TIMESTAMP NOT NULL ,  
  `IdEstilo` INT NOT NULL ,  
  `IdAmbiente` INT NOT NULL ,  
  `CalculoEstilo` INT NOT NULL ,  
  `Limited` INT NOT NULL ,  
  `BloqueoTemas` INT NOT NULL ,  
  PRIMARY KEY (`idLogConfiguracion`) )  
ENGINE = InnoDB;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

Ejemplo del funcionamiento del algoritmo paso a paso

Información de la configuración de la emisora

Estilo: 16 Ambiente: 5 Calculo: 5 Limited: 5 Bloqueotemas: 20

Id tema que suena: 56

Log Emisión: 64

Buscamos info del tema que se emite: 56

Valores del tema actual									
Estilo actual:	16	ambiente actual:	6	artista actual:	52	Votos positivos:	197	Votos negativos:	106

Buscamos características asociadas a la emisora

Valores Afines estilo

Afinidad1: 6 Afinidad2: 8 Afinidad3: 17

Valores Afines ambiente

Afinidad1: 6 Afinidad2: 10

Buscamos características asociadas al tema actual

Valores Afines estilo tema actual

Afinidad1: 6 Afinidad2: 8 Afinidad3: 17

Valores Afines ambiente tema actual

Afinidad1: 2 Afinidad2: 5

Llegamos a la selección de los temas candidatos

El tema candidato con id: 55 tiene los siguientes valores

Valor de estilo:	0	Valor de ambiente:	1	Valor de afinidad grupos:	0	Factor de novedad:	1	Distancia respecto origen:	0.45
Tema candidato:	55	Distancia candidata:	0.45						

El tema candidato con id: 42 tiene los siguientes valores

Valor de estilo:	1	Valor de ambiente:	0	Valor de afinidad grupos:	1	Factor de novedad:	1	Distancia respecto origen:	0.55
Tema	55	Distancia	0.45						

candidato: candidata:

El tema candidato con id: 40 tiene los siguientes valores

Valor de estilo:	1	Valor de ambiente:	0	Valor de afinidad grupos:	1	Factor de novedad:	Distancia 1 respecto origen:	0.55
Tema candidato:	55	Distancia candidata:	0.45					

El tema candidato con id: 39 tiene los siguientes valores

Valor de estilo:	0	Valor de ambiente:	1	Valor de afinidad grupos:	1	Factor de novedad:	Distancia 1 respecto origen:	0.55
Tema candidato:	55	Distancia candidata:	0.45					

El tema candidato con id: 38 tiene los siguientes valores

Valor de estilo:	1	Valor de ambiente:	0	Valor de afinidad grupos:	1	Factor de novedad:	Distancia 1 respecto origen:	0.55
Tema candidato:	55	Distancia candidata:	0.45					

Asignación de votos asociados

Aumentamos el estilo directo en la tabla de estilos

Aumentamos los estilos afines en la tabla de estilos

Aumentamos el ambiente directo en la tabla de ambientes

Aumentamos los ambientes afines en la tabla de ambientes

Aumentamos los votos de los temas con el mismo estilo directo

Aumentamos los votos de los temas con el mismo ambiente directo

Toca recalcular configuración del sistema

Desbloqueamos temas que correspondan, y les damos valor medio en sus votos

Tema seleccionado: 55

Se ha introducido el tema en el log de emisión

Este proyecto propone crear un sistema inteligente y autónomo de emisión conjunta por Internet y FM. A través de una interfaz Web cómodo y sencillo, el oyente podrá interactuar con la emisora e influir en la emisión, exponiendo sus preferencias. El sistema tomará de forma autónoma y dinámica las decisiones pertinentes que garantizarán que la emisión cumple las preferencias definidas.

Aquest projecte proposa crear un sistema intel·ligent i autònom de emissió conjunta per Internet i FM. A través d'una interfície Web còmoda i senzilla, l'oient podrà interactuar amb l'emissora, exposant les seves preferències. El sistema pendrà de manera autònoma i dinàmica les decisions pertinents que garanteixin que l'emissió compleixi les preferències definides

This project attempts to create an intelligent and autonomous emission system of both the Net and FM together. Through a handy and simple interface any hearer will be able to interact with the station and participate on the emission, expressing their own preferences. Such a system will make the appropriate decisions autonomously and dynamically, in such a way that will guarantee that the emission meets the previously defined preferences.